AraProp at WANLP 2022 Shared Task: Leveraging Pre-Trained Language Models for Arabic Propaganda Detection

Gaurav Singh
Independent Research
gauravsingh141116@gmail.com

Abstract

This paper presents our approach taken for the shared task on Propaganda Detection in Arabic at the Seventh Arabic Natural Language Processing Workshop (WANLP 2022). We participated in Sub-task 1, where the text of a tweet is provided, and the goal is to identify the different propaganda techniques used in it. This problem belongs to multi-label classification. For our solution, we leveraged different transformer-based pre-trained language models with fine-tuning to solve this problem. In our analysis, we found that MARBERTv2 outperforms in terms of performance, where macro-F1 is 0.08175 and micro-F1 is 0.61116 compared to other language models that we considered. Our method achieved rank 4 in the testing phase of the challenge.

1 Introduction

Two thirds of EU citizens say they see false news at least once per week (Commission et al., 2018). Propaganda, misinformation, and fake news have the power to polarise public opinion, to encourage hate speech and violent extremism, and ultimately to weaken democracies. In general terms, the spread of propaganda can be harmful to a nation and can hurt the sentiments of its people in a negative way. Currently, propaganda (or persuasion) techniques have been commonly used on social media to manipulate or mislead social media users.

There are instances where propaganda is used to divert attention from important issues by passing on fake and irrelevant information. Propaganda introduces prejudice, by hiding the other side of things, proving them wrong by introducing an element of hypocrisy rather than by logically analyzing the facts. In a similar fashion, propaganda can also hamper the critical analysis of things and stop any meaningful discussion. Some of the techniques by which propaganda is spread are loaded language, name calling, repetition, exaggeration/minimization, flag waving and many others. A detailed analysis of the other forms in which propaganda is spread is given by (Da San Martino et al., 2019). Since there are many forms through which propaganda can be spread, its detection requires a deeper analysis of the context in which the statement is made, rather than by directly labelling the whole document as propagandistic. The goal of the shared task is to build models for identifying such techniques in the Arabic social media text (specifically Tweets).

In the the shared task of Propaganda Detection in Arabic at WANLP 2022 (Alam et al., 2022), it consists of two subtasks (optional):

Subtask 1: Given the text of a tweet, identify the propaganda techniques used in it (multi-label classification problem).

Subtask 2: Given the text of a tweet, identify the propaganda techniques used in it together with the span(s) of text in which each propaganda technique appears. This is a sequence tagging task.

We participated in Subtask 1 of the same. We fine-tuned the pre-trained language models to predict the propaganda techniques for the given sentences. This is multi-label classification where more than one class can be present for identifying the sentence. We considered two multilingual language models and six Arabic language specific transformer (Vaswani et al., 2017) based language models for our analysis. We found that MARBERTv2 outperforms all other models for the specific designed experiment settings.

2 Related Work

The identification of propaganda was mainly at the level of articles. Rashkin et al. (2017) created a corpus of news articles, which were divided into four categories: propaganda, trusted, hoax, or satire. Articles from eight sources were included, two of which are propagandistic. In another work by (Da San Martino et al., 2019), they introduced a

Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP), pages 496 - 500

novel task by performing fine-grained analysis of texts by detecting all fragments that contain propaganda techniques as well as their type. There were eighteen propaganda techniques described from a novel corpus of news articles manually annotated at the fragment level. Dimitrov et al. (2021a) proposed a new multi-label multimodal task for detecting propaganda techniques used in memes from a carefully annotated new corpus of 950 memes with 22 propaganda techniques in text, image, or both. In addition, a shared task for detecting persuasive techniques in text and images was introduced at SemEval 2021. (Dimitrov et al., 2021b).

3 Data

The data of subtask 1 consists of ids, text, and propaganda techniques as labels. An example is provided in Figure 1. In our investigation, we found only 18 out of 21 classes annotated in the list of techniques provided by the organizer given in the training data. Most frequently occurring class is Loaded Language (32.8%), followed by Name Calling/Labeling, no techniques, Smears and Appeal to fear/prejudice and rest of the classes (20.3%), can be seen in the Figure 2. For training the system, we used the same training, development, and test data as provided by the organizer and the split of the data is given in Table 1.


```
{
  "id": "1389927866356412416",
  "text": "\"جدل وسخرية من ..\"\ده مش معتقل ده أحسن من اللوكاندة
    #مصر زيارات تنظمها وزارة الداخلية للسجون https://t.co
    /VkkCrRjOCF",
  "labels": [
    "Exaggeration/Minimisation",
    "Smears"
  ]
}
```


Figure 1: An sample data format from given data for subtask 1.

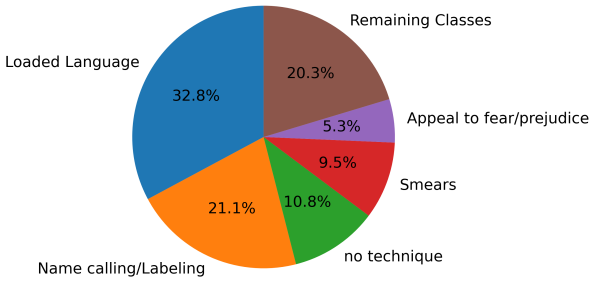

Figure 2: Class distribution in the training data including no technique.

Set	Number of Sample
Train	504
Development	52
Test	323

Table 1: Split of data provided by the organizer.

4 System Description

4.1 Model Description

In this work, we have used pre-trained transformer-based language models to identify the propaganda techniques in the sentences. Firstly, tokenized inputs were prepared based on the transformer-based language model's tokenizer for the given text, and then passed through the model, which produces contextualized word embeddings for all input tokens in our text. As we want a fixed-sized output representation, we need a pooling layer—several options like mean-pooling, max-pooling, min-pooling and many others. We simply average all contextualised word embeddings models by taking attention mask into account for correct averaging. Then, after a dropout layer was added, with a dropout rate of 0.3, we used stable dropout from the huggingface library because it is an optimised dropout model for stabilizing the training. A linear layer was added for projection into the prediction space based on the number of output classes, and a sigmoid activation function was added to each neuron output because we are dealing with multi-label classification problems.

We investigated multiple transformer-based language models, consisting of 2 multilingual models and 6 models specific to the Arabic language. A general overview showing the model architecture that we designed is depicted in Figure 3. We briefly describe the different transformer-based language models that we considered for Arabic propaganda detection.

bert-base-multilingual-cased: BERT (Devlin et al., 2018) is a transformer model pre-trained on a large corpus of multilingual data in a self-supervised fashion. A multilingual (mBERT) (Devlin et al., 2018) is a multilingual version of BERT. This model is case sensitive. It is pre-trained on the top 104 languages with the largest Wikipedia using a masked language modeling (MLM) objective.

xlm-roberta-base: RoBERTa (Liu et al., 2019) is a transformers model that was self-supervised pre-trained on a huge corpus. A multilingual version of RoBERTa is called XLM-RoBERTa (Con-

neau et al., 2019). 100 languages from 2.5TB of filtered Common Crawl data is used as its pre-training material.

bert-base-arabic: It is a pre-trained BERT base language model specifically designed for the Arabic language and was introduced by (Safaya et al., 2020). The pre-training procedure follows the training settings of BERT with some changes. It is trained for 3 million training steps with a batch size of 128, instead of 1 million with a batch size of 256. This model is pre-trained on ∼8.2 billion words: Arabic version of OSCAR (Ortiz Suárez et al., 2020) - filtered from Common Crawl, Recent dump of Arabic Wikipedia and, other Arabic resources which sum up to ∼95GB of text.

bert-base-arabert: AraBERT (Antoun et al.) is an Arabic pre-trained language model based on Google's BERT architecture (Devlin et al., 2018). It uses the same BERT-Base config. There is two versions of the model AraBERTv0.1 and AraBERTv1, with the difference being that AraBERTv1 uses pre-segmented text where prefixes and suffixes were split using the Farasa Segmenter (Darwish and Mubarak, 2016). We used AraBERTv1 for our task. The model is trained on 23GB of Arabic text consists of 70 million sentences with 3 billion words.

bert-base-arabertv2: This is similar to bert-base-arabert (Antoun et al.) but having few changes. The dataset consists of 77GB, equivalent to 200,095,961 lines or 8,655,948,860 words or 82,232,988,358 chars (before applying Farasa Segmentation). For the new dataset, authors added the unshuffled OSCAR corpus, after thoroughly filtering is done, to the previous dataset used in AraBERTv1 but with out the websites that authors previously crawled: OSCAR unshuffled and filtered (Ortiz Suárez et al., 2020), Arabic Wikipedia dump from 2020/09/01, the 1.5 billion words Arabic Corpus (El-Khair, 2016), the OSIAN Corpus (Zeroual et al., 2019) and, Assafir news articles. It used ∼3.5 times more data, and trained for longer.

ARBERT: ARBERT (Abdul-Mageed et al., 2021) is a large-scale pre-trained masked language model focused on Modern Standard Arabic (MSA). For training, it used the same architecture as BERT-base: 12 attention layers, each has 12 attention heads and 768 hidden dimensions, a vocabulary of 100K Word Pieces, making ∼163 million parameters. It is trained on a collection of Arabic datasets comprising 61 GB of text (6.2 billion tokens).

MARBERT: MARBERT (Abdul-Mageed et al., 2021) is a large-scale pre-trained masked language model focused on both Dialectal Arabic (DA) and MSA. Arabic has multiple varieties. To train it, randomly sampled 1 billion Arabic tweets from a large in-house dataset of about 6 billion tweets were obtained. Only considered those tweets with at least 3 Arabic words, based on character string matching, regardless of whether the tweet has a non-Arabic string or not. That is, authors did not remove non-Arabic so long as the tweet meets the 3 Arabic word criterion. The dataset makes up 128 GB of text (15.6 billion tokens). The same network architecture as ARBERT (BERT-base) is used, but without the next sentence prediction (NSP) objective since tweets are short.

MARBERTv2: From the results of ARBERT and MARBERT, they are not competitive on QA tasks. This can be because the two models are pre-trained with a sequence length of only 128, which does not allow them to sufficiently capture both a question and its likely answer within the same sequence window during the pre-training. To solve this problem, the authors further pre-train MARBERT on the same MSA data as ARBERT in addition to the AraNews dataset, but with a bigger sequence length of 512 tokens for 40 epochs. This pre-trained model called MARBERTv2 (Abdul-Mageed et al., 2021), to be noted it has 29 billion tokens.

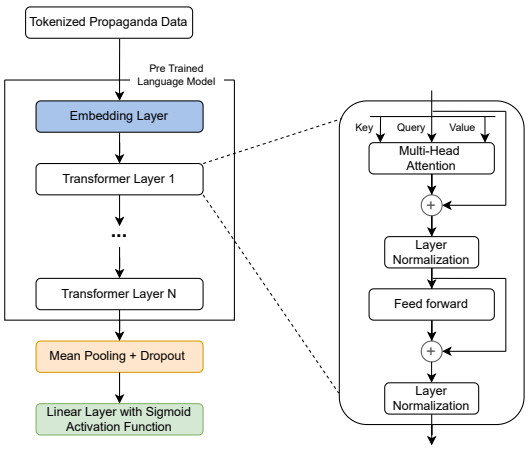

Figure 3: Fine-tuned model architecture with components built on the top of language model.

4.2 Experiment Settings

For our system, we fine-tuned the model architectures as discussed in Section 4.1. We used the AdamW (Loshchilov and Hutter, 2017) optimizer,

and binary cross entropy has been used on the output layer. The system uses the same dataset as provided by the organizer. No other data has been used. There is no extra pre-training of language models that has been done. We did not apply any extra preprocessing to the text; we simply passed the full text to the tokenizer to create tokenized inputs for the model. We have provided metric scores as provided by the challenge's portal, i.e., macro-F1 and micro-F1. All the parameters, hyper-parameters and configurations are explained in Table 2. We used the Google Colab platform for training our system, which has 12.68 GB of RAM, a 14.75 GB NVIDIA Tesla T4 GPU, and Python language. Pytorch and the Huggingface library have been used for the implementation of the system.

Parameters	Values
Epoch	10
Learning Rate	5e-5
Weight Decay	1e-2
Batch Size	4
Max Length	64
Dropout Rate	0.3
Optimizer	AdamW
Activation Function	Sigmoid
Loss Function	Binary Cross Entropy

Table 2: Parameters used for training the system.

5 Results and Discussion

In Table 3, we scored the best macro-F1 score in the bert-base-arabic model, i.e., 0.16182, and the best micro-F1 score in the MARBERTv2 model, i.e., 0.61116. The performance analysis was done after the testing phase was completed. From a challenge perspective, micro-F1 is the official metric for scoring the submission. On that basis, the MARBERTv2 model outperforms all other models. The submitted result to the challenge portal during the testing phase is for the MARBERTv2 model, where we scored 0.600 as a micro-F1 score (see Table 4).

By carefully investigating Table 3, we can observe that the range of macro-F1 scores (minimum for bert-base-arabert and maximum for bert-base-arabic, with a range of 0.09527) is approximately three times the range of micro-F1 scores (minimum for mBERT-cased and maximum for MARBERTv2, with a range of 0.0389). Our hypothesis is that it is because of the highly unbalanced class distribution where about 5 classes constitute of 80% of all the labels and the rest of 20% labels are contributed by 13 classes.

Model	macro-F1	micro-F1
mBERT-cased	0.08468	0.57226
xlm-roberta-base	0.07632	0.59186
bert-base-arabic	**0.16182**	0.59735
bert-base-arabert	0.06655	0.59222
bert-base-arabertv2	0.09965	0.60140
ARBERT	0.13366	0.60448
MARBERT	0.06969	0.60343
MARBERTv2	0.08175	**0.61116**

Table 3: Performance scores of fine-tuned language models on testing data. Here, bert-base-multilingual-cased model referred as mBERT-cased.

Model	macro-F1	micro-F1
MARBERTv2	**0.105**	**0.600**

Table 4: Submitted model result from challenge portal in testing phase.

We understand that our approach is only applicable to more general aspects of Arabic propaganda detection. Further layers must be added to the setup to capture more specific knowledge about propaganda detection in the Arabic language specific to the given dataset.

6 Conclusion

In this work, our objective is to evaluate the performance of different transformer-based language models that are being built with simple fine-tuning. In the course of doing this, we achieved rank 4 on the challenge leaderboard without explicitly adding additional processing. We understand that propaganda detection is a challenging task. Our approach sets the baseline for the general aspects of Arabic propaganda detection. For future work, we can apply data augmentation, cross-validation, an ensemble of models, and further fine-tuning of model architecture specific to the task.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. Arbert amp; marbert: Deep bidirectional transformers for arabic.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Preslav Nakov, and Giovanni Da San Martino. 2022.

Overview of the WANLP 2022 shared task on propaganda detection in Arabic. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop*, Abu Dhabi, UAE. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

European Commission, Content Directorate-General for Communications Networks, and Technology. 2018. *Fake news and disinformation online*. Publications Office of the European Union.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics.

Kareem Darwish and Hamdy Mubarak. 2016. Farasa: A new fast and accurate Arabic word segmenter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1070–1074, Portorož, Slovenia. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021a. Detecting propaganda techniques in memes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6603–6617, Online. Association for Computational Linguistics.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021b. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Ibrahim Abu El-Khair. 2016. 1.5 billion words arabic corpus. *ArXiv*, abs/1611.04033.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark. Association for Computational Linguistics.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182, Florence, Italy. Association for Computational Linguistics.